Luke Scott

COSC 311

Lab 2

Dr. Wang

**Lab 2 Lab Report and Source Code**

**Task 1:**

.info()

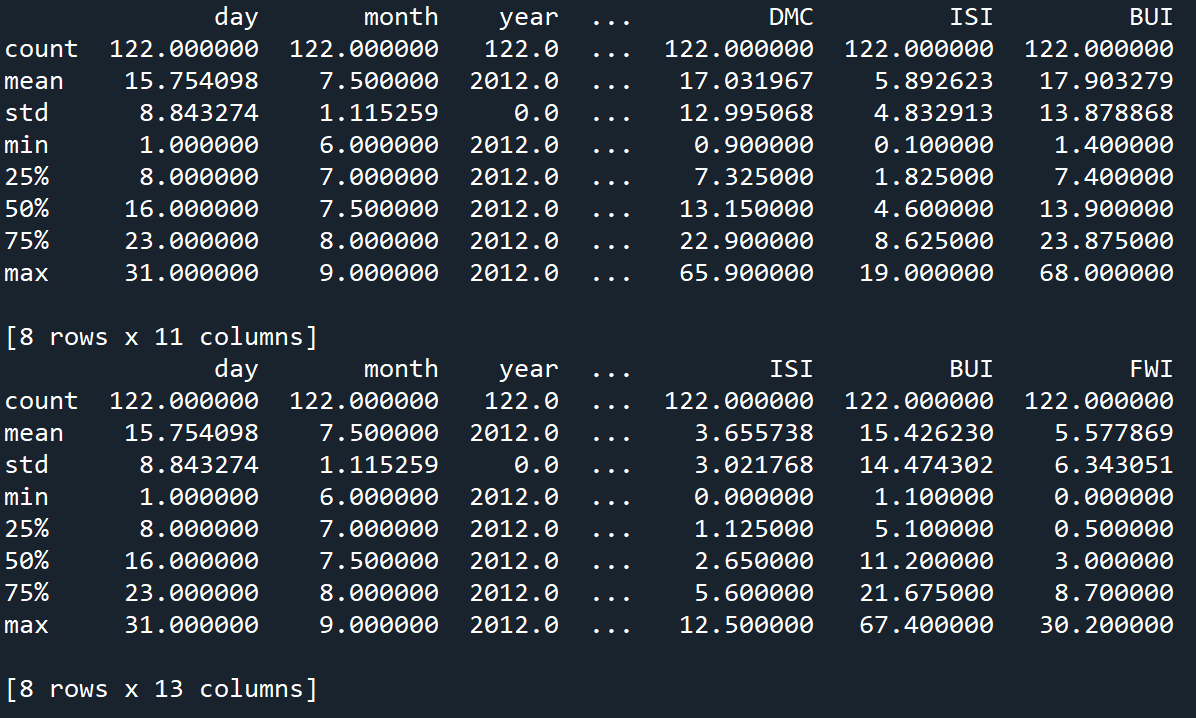
A picture containing text, plaque

Description automatically generated

A picture containing text, plaque

Description automatically generated

.describe()



[‘ Ws’]

Text

Description automatically generated

.count()

Text

Description automatically generated

**Task 2:**

Chart, line chart

Description automatically generated

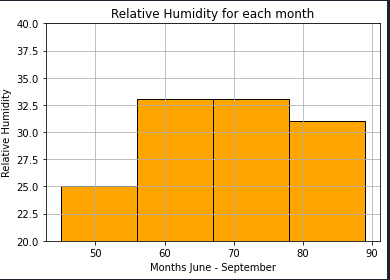
**Task 3:**

**Chart, scatter chart

Description automatically generated**

**Task 4:**

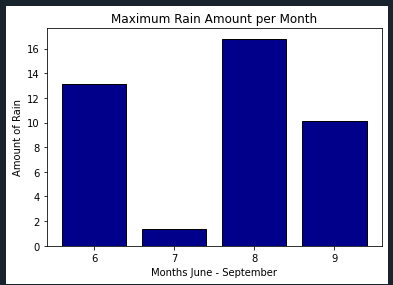
**--- NOTE :** I was experimenting with matplotlib and wasn’t happy with the histogram result for this task. I included a bar graph of the data as well, this is commented out from lines 52-61

****

**Chart, bar chart

Description automatically generated**

**Task 5:**

****

**Task 6:**

**Chart, histogram

Description automatically generated**

**Task 7:**

**Chart, line chart, histogram

Description automatically generated**

**Task 8:**

**Chart, histogram

Description automatically generated**

**Task 9:**

**Chart, bar chart

Description automatically generated**

**Source Code:**

import pandas as pd

from matplotlib import pyplot as plt

from collections import Counter

SBA\_Region = pd.read\_csv('Sidi-Bel\_Abbes\_Region.csv')

B\_Region = pd.read\_csv('Bejaia\_Region.csv')

# Task 1 : Show Info(), Describe(), 'Ws', count() of both data sets

SBA\_Region.info()

B\_Region.info()

B\_DF = pd.DataFrame(B\_Region)

SBA\_DF = pd.DataFrame(SBA\_Region)

print(SBA\_Region.describe(), '\n')

print(B\_Region.describe(), '\n')

print(SBA\_Region[' Ws'], '\n')

print(B\_Region[' Ws'], '\n')

print(SBA\_Region.count(), '\n')

print(B\_Region.count(), '\n')

# Gathers the mean of all temperatures based on month

temperature\_mean = B\_Region.groupby('month')[['Temperature']].mean().sort\_values('month')

# Creates a list of integers referencing months from the data set

months = Counter(B\_Region['month'])

# Task 2 = Show line graph of average temperature change by month

plt.plot(months.keys(), temperature\_mean, color='green', marker='o', linestyle='solid')

plt.title('Temperature Change by Month in Bejaia Region')

plt.ylabel('Temperature (Celcius)')

plt.xlabel('Months June - September')

plt.xticks([6, 7, 8, 9])

plt.show()

# Task 3 = Create scatter plot for temperature based on Fire Weather Index

SBA\_Region.plot.scatter(x = 'Temperature', y = 'FWI')

plt.show()

# Task 4 = Draw a historgram of average Relative Humidity by month

B\_DF[' RH'].plot(kind = 'hist', bins = 4, color = 'orange', edgecolor = 'black')

plt.grid(visible = True)

plt.title('Relative Humidity for each month')

plt.xlabel('Months June - September')

plt.ylabel('Relative Humidity')

plt.ylim(20, 40)

plt.show()

'''

b\_rh = B\_DF.groupby('month')[' RH'].mean()

plt.bar(b\_rh.keys(), b\_rh, color = 'orange', edgecolor = 'black')

plt.title('Relative Humidity for each month')

plt.xlabel('Months June - September')

plt.ylabel('Relative Humidity')

plt.xticks([6,7,8,9])

plt.ylim(50, 75)

plt.show()

'''

# Task 5 = Draw a bar graph to show maximum rain amount for each month

b\_max\_rain = B\_DF.groupby('month')['Rain '].max()

plt.bar(b\_max\_rain.keys(), b\_max\_rain, color = 'darkblue', edgecolor = 'black')

plt.title('Maximum Rain Amount per Month')

plt.xlabel('Months June - September')

plt.ylabel('Amount of Rain')

plt.xticks([6, 7, 8, 9])

plt.show()

# Task 6 = Draw a histogram of Windspeed in the month of June

(SBA\_DF[' Ws'][SBA\_DF['month'] == 6]).hist(bins = 5, color = 'lightblue', edgecolor = 'black')

plt.show()

# Task 7 = Draw a line figure that shows correlation between Temperature and Relative Humidity

SBA\_Temp = (SBA\_DF[' RH'][SBA\_DF['month'] == 7])

SBA\_RH = (SBA\_DF['Temperature'][SBA\_DF['month'] == 7])

xs = [i for i, \_ in enumerate(SBA\_Temp)]

plt.plot(xs, SBA\_Temp, 'g-', label = 'Temperature')

plt.plot(xs, SBA\_RH, 'r-.', label = 'Rain Humidity')

plt.xticks([])

plt.title('Rain Hmidity and Temperature Correlation')

plt.show()

# Task 8 = Draw a bar graph to show the distribution of Relative Humidity

values = Counter(min(humid // 10 \* 10, 90) for humid in B\_DF[' RH'])

plt.bar([x + 5 for x in values.keys()], values.values(), 10, edgecolor = 'black', color = 'red')

plt.axis([35, 95, 0, 45])

plt.xticks([10 \* i for i in range(4, 10)])

plt.xlabel('Decile')

plt.ylabel('Number of Days')

plt.title('Distribution of Relative Humidity')

plt.show()

# Task 9 = Draw a figure to show average temperature for each month with fires and no fires

fire = B\_DF.groupby(['month', 'Classes '])['Temperature'].mean()

fire.plot(x = 'month', y = 'Temperature', kind = 'bar',\

color = ['orange','lightgreen'], edgecolor = 'black', ylim=(25,35),\

xlabel = 'Months by Fire', ylabel = 'Temperature', title = 'Average Temperature By Month Based on Fires')